Full credit is given to the above companies including the OS that this PDF file was generated!

PowerShell Get-Help on command 'Install-PackageProvider'

PS C:\Users\wahid> Get-Help Install-PackageProvider

```
NAME
    Install-PackageProvider


SYNOPSIS
    Installs one or more Package Management package providers.



SYNTAX
    Install-PackageProvider [-Name] <System.String[]> [-AllVersions] [-Credential
    <System.Management.Automation.PSCredential>] [-Force] [-ForceBootstrap]
    [-MaximumVersion <System.String>] [-MinimumVersion <System.String>] [-Proxy
    <System.Uri>] [-ProxyCredential <System.Management.Automation.PSCredential>]
    [-RequiredVersion <System.String>] [-Scope {CurrentUser | AllUsers}] [-Source
    <System.String[]>] [-Confirm] [-WhatIf] [<CommonParameters>]


    Install-PackageProvider [-InputObject]
    <Microsoft.PackageManagement.Packaging.SoftwareIdentity[]> [-AllVersions]
    [-Force] [-ForceBootstrap] [-Proxy <System.Uri>] [-ProxyCredential
    <System.Management.Automation.PSCredential>] [-Scope {CurrentUser | AllUsers}]
    [-Confirm] [-WhatIf] [<CommonParameters>]
```

DESCRIPTION

The `Install-PackageProvider` cmdlet installs matching Package Management providers that are available in package sources registered with PowerShellGet . By default, this includes modules available in the Windows PowerShell Gallery with the PackageManagement tag. The PowerShellGet Package Management provider is used for finding providers in these repositories.

This cmdlet also installs matching Package Management providers that are available using the Package Management bootstrapping application.

This cmdlet also installs matching Package Management providers that are available in the Package Management Azure Blob store. Use the bootstrapper provider to find and install them.

In order to execute the first time, PackageManagement requires an internet connection to download the NuGet package provider. However, if your computer does not have an internet connection and you need to use the NuGet or PowerShellGet provider, you can download them on another computer and copy them to your target computer. Use the following steps to do this:

1. Run `Install-PackageProvider -Name NuGet -RequiredVersion 2.8.5.201 -Force` to install the provider from a computer with an internet connection. 1. After the install, you can find the provider installed in `$env:ProgramFiles \PackageManagement\ProviderAssemblies<ProviderName><ProviderVersion>` or `$e nv:LOCALAPPDATA\PackageManagement\ProviderAssemblies<ProviderName><ProviderVers ion>`. 1. Place the `<ProviderName>` folder, which in this case is the NuGet folder, in the corresponding location on your target computer. If your target computer is a Nano server, you need to run `Install-PackageProvider` from Nano Server to download the correct NuGet binaries. 1. Restart PowerShell to auto-load the package provider. Alternatively, run `Get-PackageProvider -ListAvailable` to list all the package providers available on the computer. Then use `Import-PackageProvider -Name NuGet -RequiredVersion 2.8.5.201` to

```
    import the provider    to the current Windows PowerShell session.


PARAMETERS
    -AllVersions <System.Management.Automation.SwitchParameter>
        Indicates that this cmdlet installs all available versions of the package
        provider. By default, `Install-PackageProvider` only returns the highest
        available version.

    -Credential <System.Management.Automation.PSCredential>
        Specifies a user account that has permission to install package providers.

    -Force <System.Management.Automation.SwitchParameter>
        Indicates that this cmdlet forces all actions with this cmdlet that can be
        forced. Currently, this means the Force parameter acts the same as the
        ForceBootstrap parameter.

    -ForceBootstrap <System.Management.Automation.SwitchParameter>
        Indicates that this cmdlet automatically installs the package provider.

    -InputObject <Microsoft.PackageManagement.Packaging.SoftwareIdentity[]>
        Specifies a SoftwareIdentity object. Use the `Find-PackageProvider` cmdlet
        to obtain a SoftwareIdentity object to pipe into `Install-PackageProvider`.

    -MaximumVersion <System.String>
        Specifies the maximum allowed version of the package provider that you
        want to install. If you do not add this parameter,
        `Install-PackageProvider` installs the highest available version of the
        provider.

    -MinimumVersion <System.String>
        Specifies the minimum allowed version of the package provider that you
        want to install. If you do not add this parameter,
```

`Install-PackageProvider` installs the highest available version of the package that also satisfies any requirement specified by the MaximumVersion parameter.

-Name <System.String[]>

Specifies one or more package provider module names. Separate multiple package names with commas. Wildcard characters are not supported.

-Proxy <System.Uri>

Specifies a proxy server for the request, rather than connecting directly to the Internet resource.

-ProxyCredential <System.Management.Automation.PSCredential>

Specifies a user account that has permission to use the proxy server that is specified by the Proxy parameter.

-RequiredVersion <System.String>

Specifies the exact allowed version of the package provider that you want to install. If you do not add this parameter, `Install-PackageProvider` installs the highest available version of the provider that also satisfies any maximum version specified by the MaximumVersion parameter.

-Scope <System.String>

Specifies the installation scope of the provider. The acceptable values for this parameter are:

- AllUsers - installs providers in a location that is accessible to all users of the computer.   By default, this is $env:ProgramFiles\PackageManagement\ProviderAssemblies. - CurrentUser - installs providers in a location where they are only accessible to the current   user. By default, this is $env:LOCALAPPDATA\PackageManagement\ProviderAssemblies.

-Source <System.String[]>

    Specifies one or more package sources. Use the `Get-PackageSource` cmdlet
    to get a list of available package sources.

-Confirm <System.Management.Automation.SwitchParameter>

    Prompts you for confirmation before running the cmdlet.

-WhatIf <System.Management.Automation.SwitchParameter>

    Shows what would happen if the cmdlet runs. The cmdlet is not run.

<CommonParameters>

    This cmdlet supports the common parameters: Verbose, Debug,
    ErrorAction, ErrorVariable, WarningAction, WarningVariable,
    OutBuffer, PipelineVariable, and OutVariable. For more information, see
    about_CommonParameters (https:/go.microsoft.com/fwlink/?LinkID=113216).

Example 1: Install a package provider from the PowerShell Gallery

```
Install-PackageProvider -Name "GistProvider" -Verbose
```

- Example 2: Install a specified version of a package provider -

```
Find-PackageProvider -Name "NuGet" -AllVersions
Install-PackageProvider -Name "NuGet" -RequiredVersion "2.8.5.216" -Force
```

---------- Example 3: Find a provider and install it ----------

```
Find-PackageProvider -Name "GistProvider" | Install-PackageProvider -Verbose
```

Example 4: Install a provider to the current user's module folder

```
Install-PackageProvider -Name GistProvider -Verbose -Scope CurrentUser
```

REMARKS

To see the examples, type: "get-help Install-PackageProvider -examples".

For more information, type: "get-help Install-PackageProvider -detailed".

For technical information, type: "get-help Install-PackageProvider -full".

For online help, type: "get-help Install-PackageProvider -online"